ARMY RESEARCH LABORATORY

Toward Autonomous Multi-floor Exploration: Ascending Stairway Localization and Modeling

by Jeffrey A. Delmerico, Jason J. Corso, David Baran, and Philip David

ARL-TR-6381 **March 2013**

NOTICES

Disclaimers

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-6381 **March 2013**

Toward Autonomous Multi-floor Exploration: Ascending Stairway Localization and Modeling

Jeffrey A. Delmerico, Jason J. Corso, David Baran, and Philip David
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
March 2013	Final	

4. TITLE AND SUBTITLE Toward Autonomous Multi-Floor Exploration: Ascending Stairway Localization and Modeling	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Jeffrey A. Delmerico, Jason J. Corso, David Baran, and Philip David	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-CII-A 2800 Powder Mill Road Adelphi, MD 20783-1197	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-6381
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

primary author’s email: <david.g.baran.civ@mail.mil>

14. ABSTRACT

Localization and modeling of stairways by mobile robots can enable multi-floor exploration for those platforms capable of stair traversal. No system yet presented is capable of localizing a stairway on a map and estimating its properties, two functions that would enable stairways to be considered as traversable terrain in a path planning algorithm. We propose a system for detecting and modeling an ascending stairway while performing simultaneous localization and mapping. We design a generative model of a stairway as a single object and localize it with respect to the map, as well as estimate the dimensions of its steps. Modeling the stairway as a whole will enable exploration of higher floors of a building by allowing the stairway to be incorporated into path planning by considering it as a portal to new frontiers. Our system consists of two parts: a computationally efficient detector that leverages geometric cues from depth imagery to detect sets of ascending stairs, and a stairway modeler that uses multiple detections to infer the location and parameters of a stairway that is discovered during exploration. We demonstrate the performance of this system when deployed on several mobile platforms.

15. SUBJECT TERMS

Stair detection, stairway modeling, autonomous exploration, unmanned ground vehicle

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					David Baran
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	32	**19b. TELEPHONE NUMBER** (*Include area code*)
Unclassified	Unclassified	Unclassified			(301) 394-5613

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Acknowledgment

This material is based upon work partially supported by the Federal Highway Administration (FHWA) under Cooperative Agreement No. DTFH61-07-H-00023, the Army Research Office (ARO) (W911NF-11-1-0090) and the National Science Foundation (NSF) CAREER grant (IIS-0845282). Any opinions, findings, conclusions or recommendations are those of the authors and do not necessarily reflect the views of the FHWA, ARO, or NSF.

INTENTIONALLY LEFT BLANK.

1. Introduction

Autonomous mobile robots have traditionally been restricted to single floors of a building or outdoor areas free of abrupt elevation changes such as curbs and stairs. This restriction presents a significant limitation to real-world applications, such as whole-building mapping and rescue scenarios. Our work seeks a solution to this problem and is motivated by the rich potential of an autonomous ground robot that can climb stairs while exploring a multi-floor building. Our proposed solution to this problem is a system to detect and localize stairways in the environment during the process of exploration, and model any identified stairways in order to determine if they are traversable by the robot. With a map of the environment and estimated locations and parameters of the stairways, the robot could plan a path that traverses the stairs in order to explore the frontier at other elevations that were previously inaccessible. For example, a robot could finish mapping the ground floor of a building, return to a stairway that it had previously discovered, and ascend to the second floor to continue exploring if that stairway is of dimensions (i.e., step height, width, and pitch) that are traversable by that particular platform. Autonomous multi-floor exploration is a new behavior for ground robots, and we present this work as a first step toward the realization of that capability.

Other systems have been proposed for related tasks, but no existing work approaches the problem in the context of the aforementioned scenario. Several methods (*1–3*) perform detection and traversal, but do not model the pose or location of the stairway and simply immediately initiate a climbing mechanism. Although these capabilities are related to our problem scenario, immediate climbing is not necessarily compatible with a mapping task. Path planning for multi-floor exploration should take the stairway into account as a portal to more unexplored regions, but traversing stairs immediately upon a single detection makes exploring the low-cost frontiers of the original floor more difficult and may fail if the detection was erroneous. Several other existing approaches (*4–6*) perform modeling of individual steps, or sets of them, but on a level that is both unnecessarily detailed for the purposes of localization and too computationally expensive to be practical on a platform whose primary task is exploration, not stairway detection. In particular, to determine if a stairway is traversable and locate it on a map, the individual steps and risers do not need to be modeled as planes nor does all of their corresponding three-dimensional (3-D) data need to be extracted.

Our proposed system directly addresses the needs of an exploratory platform for solving the problem defined above. It is composed of a stairway detection module for extracting stair edge points in 3-D from depth imagery and a stairway modeling module that aggregates many such detections into a single point cloud from which the stairway's dimensions and location are estimated (figure 1). Modeling the stairway over many detections allows the system to form a complete model from many partial observations. We model the stairway as a single object: an inclined plane constrained by a bounding box, with stair edges lying in the plane. As new observations are added to the aggregated point cloud, the model is re-estimated, outliers are removed, and well-supported stair edges are used to infer the dimensions of each step. Section 3 contains a more in-depth description of our approach.

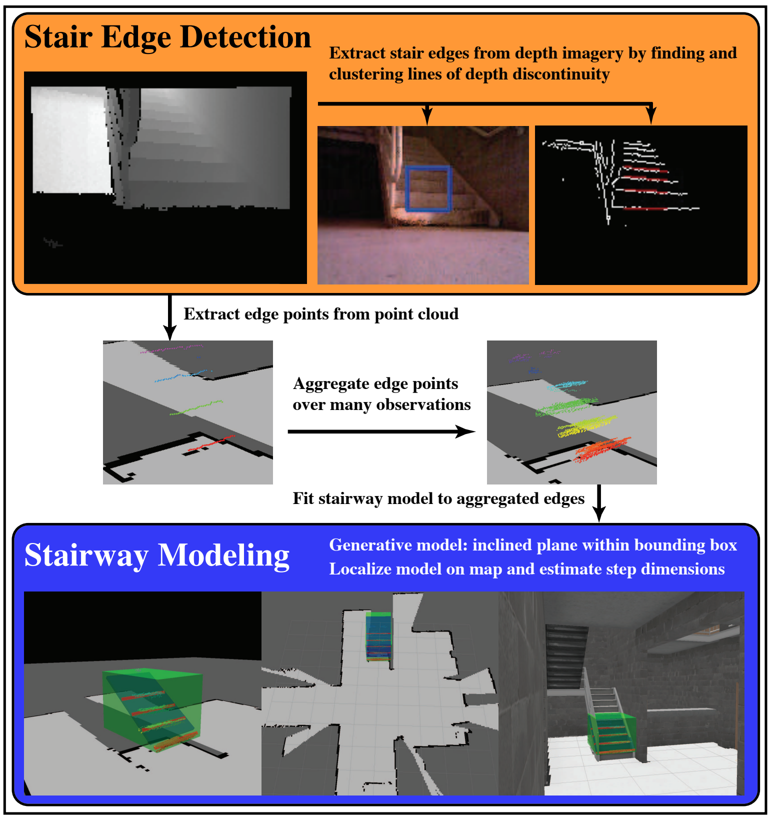

Figure 1. High level workflow of the proposed system, consisting of two modules: stair edge detection and stairway modeling. Stair edges are extracted from depth imagery and collected over many observations into an aggregated point cloud. Periodically, a generative model of a stairway is fit to the aggregate cloud and its parameters re-estimated. The result is a model localized with respect to the robot's map of its environment. (Data: Building 1 Front trial of the Military Operation in Urban Terrain [MOUT] dataset).

We have deployed this system on an iRobot PackBot as well as a TurtleBot, both fitted with Microsoft Kinect depth sensors. Our system runs in real time and demonstrates robust and accurate performance in both localization and parameter estimation for a wide variety of stairways (see section 4).

This report presents the following contributions:

- Initial step toward new ground robot behavior: autonomous multi-floor exploration. Locate stairwells during mapping of environment and later ascend them to explore new frontiers.
- A minimalist generative stairway model: an inclined plane constrained within a bounding box. This provides the right level of detail to determine if the stairway is traversable by the robot. If necessary, more detailed modeling can be performed in the context of subsequent stair traversal.
- Aggregation of many partial views into a coherent object model. Re-estimation and outlier removal permits estimation of a robust aggregate model in the presence of a low-recall detector and imprecise alignment.
- A novel approach to stair detection. Find lines of depth discontinuity representing convex stair edges and enforce the constraints of the depth image signature of a stairway (nearly parallel, overlapping in the vertical direction in image coordinates and lying on an inclined plane). Filter in edge points from the associated point cloud as a partial view of the stairway.

---

2. Related Work

Several existing methods perform stairway detection but immediately initialize a traversal procedure, which is not necessarily desirable in an exploration scenario. In general, these detection methods provide only a bearing, and possibly a distance, to the stairway relative to the robot's pose, and only serve to trigger the autonomous traversal phase of their systems. Ray et al. (*3*) designed a robotic platform specifically for stairway traversal that performs stair edge detection using Canny edge detection and a Hough transform on camera imagery. Their approach imposes some similar constraints to ours on the detected line segments, but they use this information only to determine a bearing to the stairway for immediate traversal. Some detection and pose correction for traversal are presented in Mihankhah et al. (*2*), where they use a vertically oriented laser range finder to detect the set of regular discontinuities corresponding to stairs and then maneuver the robot so it is lined up for traversal. Johnson et al. (*1*) detect stairs using a horizontal two-dimensional (2-D) laser scanner, but their stair climbing behavior is initiated as soon as a detection is made. However, their approach to traversal of stairs with landings is an important contribution to multi-floor exploration. Other stair edge detection systems have been proposed in the context of controller feedback for stair traversal (*7, 8*) and object detection (*9*).

Some techniques have been presented for performing modeling of steps or sets thereof, but these methods are not presented in the context of mapping and localization, and provide higher levels of detail than are necessary for a robot to find and evaluate a stairway for later traversal. Osswald et al. (*4*) use a combination of a 2-D laser range finder and a monocular camera to extract step edges so a humanoid robot can perform step planning to ascend a spiral staircase. Pradeep et al. (*5*) also use stereo vision and seek to explicitly model steps and stairs, and they perform plane fitting using local normals, Random Sample Consensus (RANSAC), and clustering by tensor voting. Lu and Manduchi (*6*) use Canny edge detection on one of the camera images from a stereo sensor, and then compute a measure of concavity/convexity on those lines in the stereo depth field in order to model curbs and stairs as alternating concave and convex lines on parallel planes. All of these methods exhibit good accuracy and robustness, but the models are unnecessarily fine-grained for localizing a stairway during exploration, and are too slow for real-time use.

The works by Hernandez and Jo (*10*) and Hernandez et al. (*11*) represent the most similar approaches to ours in terms of the goal of detecting and localizing sets of stairs, but are independent of modeling or mapping. In reference 10, they segment outdoor staircases from single monocular images using line detection and vanishing point analysis, and in reference 11 they use some of the same line techniques (Gabor filtering) along with motion stereo to detect and localize indoor stairways. In principle, their motion stereo approach could also be applied to outdoor imagery for localization. However, the scope of the work is limited to detection and a computation of bearing relative to the robot's pose.

There is some prior work in modeling stairways as whole objects, but only in an ontological context within the geoinformation literature. The work by Schmittwiken et al. in (*12*) presents a Unified Modeling Language (UML)/Object Constraint Language (OCL) grammar for modeling stairways as symmetric and partially recursive structures, while later work (*13*) demonstrates the fitting of models from this grammar to 3-D data. Although they do seek to model the staircase as a single object (that is also composed of repetitive sub-structures), like the other modeling approaches, it is more detailed and computationally intensive than is necessary for localization while mapping. However, their approach is amenable to localization of a full stairway model for urban 3-D modeling.

Some recent work in multi-floor mapping may provide some of the tools for implementing our desired system. Shen et al. (*14*) have demonstrated that multi-floor exploration is possible in open indoor environments with an unmanned aerial vehicle. Although their platform by nature avoids the need for stair detection and traversal, their approach for mapping may one day be applicable for ground vehicles. The barometric method presented by Ozkil et al. (*15*) for measuring elevation, and therefore distinguishing floors of a building, will likely also be useful in implementing our desired multi-floor exploratory system.

The most comprehensive system yet presented is also one that aims to perform the complementary task to our detection and localization of ascending stairs: detection and traversal of descending staircases. Hesch et al. (*17*) use a combination of texture, optical flow, and geometry from a monocular camera to detect candidate descending stairwells, navigate to them, and then align with and traverse them. Although they do not perform any explicit mapping of stairwell location or present quantitative accuracy results, their implementation runs in real time and the detector module from their implementation could be extracted and paired with our ascending stair detector in a comprehensive multi-floor mapping system. No system has yet been proposed for both ascending and descending stairway detection and traversal.

The related work in this area falls into three rough categories: stairway detection as a trigger for traversal (*1–3*), fine-grained modeling (*4–6*), and localization relative to the robot's pose (*10, 11*). Unlike these works, ours is the first effort to place stairway detection and localization in the context of mapping and decouple detection from traversal, which will enable multi-floor path planning. We also present a minimal model that is sufficient to describe the location and traversability of a stairway without the fine-grained modeling of the state of the art that is unnecessarily expensive on a platform whose objective is exploration.

3. Methods

3.1 Stairway Model

We propose a generative model to represent a stairway as a single object. For localization and path planning purposes, planar models for all of the steps and risers are unnecessary, so a simpler model will suffice: step dimensions and pitch should be adequate to allow the robot to determine whether a stairway is traversable, and localization on a map should enable the robot to return to ascend the stairs at a later time.

Our model consists of an inclined plane constrained by a bounding box, with stair edges wherever there are well-supported clusters in the plane (figure 2). This model is parameterized by the bounding box centroid (B_x,B_y,B_z) and dimensions (H,W,D), pitch relative to the ground plane (P), and step dimensions (h,d). We assume that stair steps are approximately parallel to the ground plane, so the bounding box top and bottom are parallel to the XY plane. For an inclined plane model of

$$ax + by + cz + d = 0, \tag{1}$$

the planes constituting the bounding box are given by

$$z = B_z \pm \frac{H}{2} \tag{2}$$

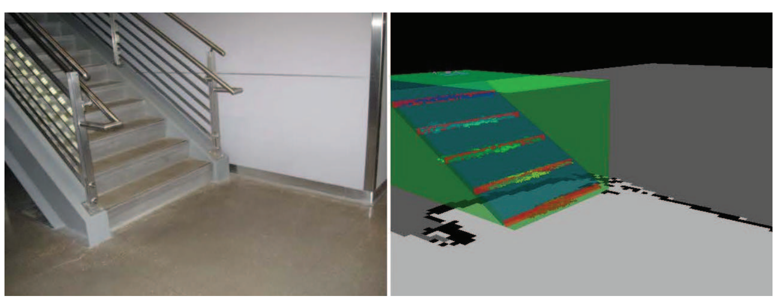

Figure 2. Stairway with corresponding model consisting of bounding box (green), planar model (blue), and step edges (red), as well as edge point cloud support for step edge lines (rainbow). (Data: Davis Hall Front trial from State University of New York [SUNY] at Buffalo [UB] dataset).

$$a(x - B_x) + b(y - B_y) \pm \frac{D}{2}\left(\sqrt{a^2 + b^2}\right) = 0 \tag{3}$$

$$-cb(x - B_x) + ca(y - B_y) \pm \frac{W}{2}\left(c\sqrt{a^2 + b^2}\right) = 0 \tag{4}$$

for the top/bottom, front/back, and sides, respectively. Here, the pitch P is computed as the dihedral angle of the planar model and the ground plane: $P = \arccos{(a, b, c)} \cdot (0, 0, 1)$. We infer the parameters of the model from the extracted points corresponding to step edges.

3.2 Localization and Modeling

In order to build up a complete model of a stairway, we piece together many incomplete views, potentially from many different perspectives, and estimate the parameters of the model from the aggregate pool of data (see algorithm 1). Starting with an empty point cloud representing the stair edges, we add to it the extracted edge points from each subsequent observation. We do not explicitly align the detected edges, but instead rely on the robot's estimated pose to approximately align the independent observations (figure 3), and implement a number of statistical techniques to ensure that the resulting model is robust to outliers and imprecise point cloud alignment. Additionally, since a misaligned detection will equally affect all of the step edges extracted from that observation, the step dimension estimates will not be subject to alignment error.

Algorithm 1 Stairway Modeling

Initialize point cloud E to be empty
for Each detection P' (see Alg. 2) **do**
  Add points from P' to E
  if # of detections is divisible by k **then**
    Downsample E to fine voxel grid
    Perform statistical outlier removal
    Fit a plane m to E using RANSAC and compute pitch relative to ground plane
    Remove outliers of m from E
    Fit bounding box B to E and compute stairway dimensions (H,W,D)
    Project E onto cross-sectional plane x, orthogonal to m and passing through bounding box centroid
    Find Euclidean clusters C from projected cloud Ep and compute their centroids
    Sort C by ascending height, and compute differences in height and depth between adjacent centroids with $> n$ points of support
    Average height and depth differences to compute step dimensions (h, d)
  end if
end for

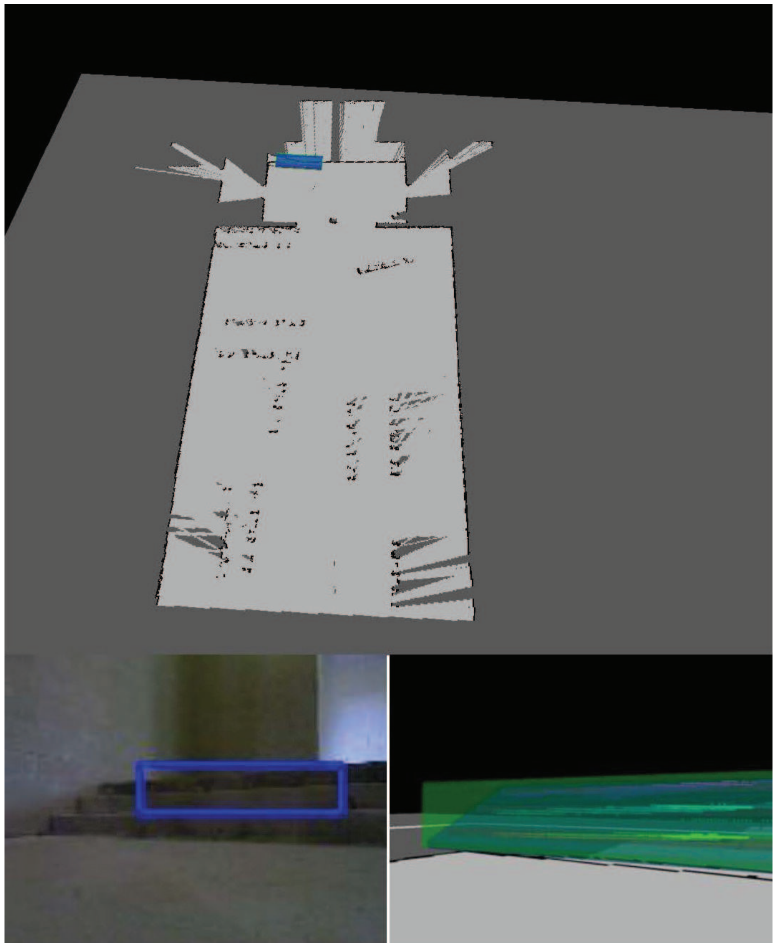

Figure 3. Figure showing a map marked up with a marker for the model of the detected stairwell, in addition to a camera image and model view from the robot's perspective. (Data: Building 7 Interior trial from MOUT dataset).

Since each observation only adds a partial view of the stairway, we periodically re-estimate the parameters of the model (in our experiments, after k = 5 or 10 observations). Our detector operates at almost the full frame rate of the camera, so more frequent modeling is redundant. We perform the following steps in order to estimate the model's parameters.

To prevent our aggregate edge point cloud E from growing without bound, we first downsample E to a 1-cm voxel grid (other grid sizes affect primarily speed and not modeling performance). We perform statistical outlier removal to reduce noise in E. To the remaining points, we fit a planar model p with RANSAC (*19*) and remove any outliers from E.

We then infer the parameters of the stairway model from the remaining points in E. We determine the bounding box centroid and dimensions by fitting a rectangular prism to the data that is aligned with the ground plane but rotated in the XY plane to match the alignment of p. We next compute the cross-sectional orthogonal plane that passes through (B_x,B_y,B_z) and project the points of E to it. When accounting for alignment errors and unequal observation of each step, we would expect there to be a cluster of projected points around each step edge. We therefore find Euclidean clusters on the projected plane and treat each well-supported cluster center (> 250 points) as a stair edge. We compute the differences in height and depth between each pair of adjacent cluster centers, and then average these differences to determine the step dimensions (h and d).

3.3 Stair Edge Detection

Inspired by some of the techniques used in other methods (*2–4, 6–8, 10*), we have developed an ascending stairway detector that exploits the geometric properties that steps display in depth images. On a deployed system, it runs in real time with high accuracy and robustness. In particular, we find lines in a depth image that represent discontinuities where the depth from the camera changes abruptly. In the depth field, a set of stairs will have a discontinuity at the edge of each step that is above the height of the sensor. The tops of lower steps will be visible in the sensor's field of view and may not exhibit a strong enough depth discontinuity to be detected as edges. Regardless of the horizontally rotated viewing angle of the camera, these discontinuities will form a set of nearly parallel lines (with some effects of perspective) for all but tight spiral staircases. We leverage this distinct depth signature by detecting all such lines of discontinuity in the image, filtering and clustering them to find a near-parallel set, and ultimately fitting a plane to the extracted stair edge points to confirm or reject the stairwell candidate hypothesis if they lie on an inclined plane of traversable angle. By detecting these lines of discontinuity in the depth field rather than a camera image, our detector is robust to appearance.

Given an input depth image, our algorithm proceeds in the following steps (please refer to algorithm 2 and figure 4 for visual reference). All parameter values in the detection module were determined empirically but remained fixed throughout all experiments.

We first perform edge detection with the Canny operator (*20*) to extract all pixels that lie on a discontinuity in the depth field into an edge image. We use parameter values of 30 and 40 for the two thresholds, and a kernel size of 3.

Algorithm 2 Stair Edge Detection

1: Input: depth image D and co-registered point cloud P provided by depth sensor
2: Perform Canny edge detection on D to produce edge image E
3: Set to zero all edge pixels of E that border a depth value of 0 in an 8-connected neighborhood
4: Generate a set of candidate lines L using the probabilistic Hough transform on E
5: Compute the slope in pixel coordinates of all lines in L, and merge any nearly collinear lines
6: Compute a histogram of the slopes in L with 10° bins, and extract lines in the bin with largest frequency $\pm 5^\circ$ into L'
7: Compute a histogram H of the number of lines passing through each column of the image
8: Find the maximum frequency in H, and find the left- and right-most columns (l and r) containing this frequency
9: Find the upper- and bottom-most lines (u and b) in the columns between l and r
10: Remove all lines from L' that do not fall within the box bounded by u, r, b, and l
11: Reject image as a positive detection if $\mid L' \mid < 3$ or $r - l < 10$ (enforce multiple steps and reasonable overlap of lines)
12: Extract the points from P corresponding to the lines in L' into a new point cloud P'
13: Fit a least-squares plane p to the points from P'
14: Reject image if the dihedral angle ($\phi = \arccos(n_p \cdot n_{horiz})$) between p and the horizontal is $> 45^\circ$
15: Return P'

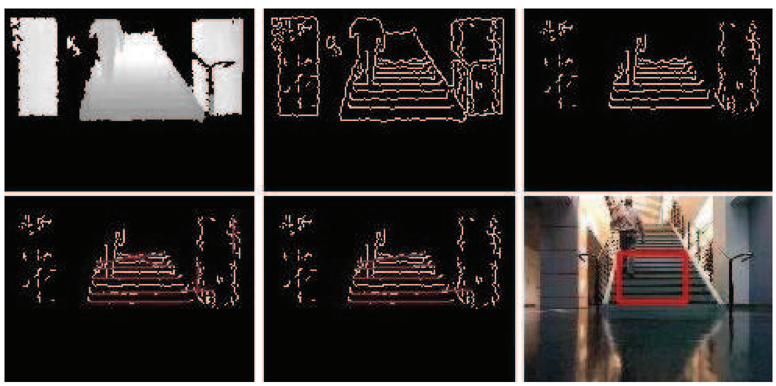

Figure 4. An example frame from an indoor testing video. Top row (L to R): source depth image, edge image, edge image with boundary lines removed. Bottom row (L to R): candidate lines (in red) before filtering for orientation and clustering, candidate lines after filtering, marked up camera image with bounding box.

Even the most robust correspondence algorithm can still fail to compute depth for some regions in an image due to poor texture, lighting effects, or occlusion. Structured light depth sensors also are unable to compute depth for every pixel due to lighting or other error sources. In a depth image, these invalid regions are often set to zero. To handle such data, we filter the initial edge image to remove all edge pixels that lie on the boundary between valid and invalid depth regions. In this way, we explicitly avoid considering any non-physical discontinuities in the depth image.

We next use a probabilistic Hough transform (*21*) to detect straight lines from the edge image with boundary lines removed. Parameter values for this step are distance and angle resolutions of 1 pixel and 1° for the accumulator, an accumulator threshold of 20, a minimum line length of 20, and a maximum gap of 5.

Since some detected lines may be nearly collinear, we consider all pairs of lines $L_i : y = m_i x + b_i$ and $L_j : y = m_j x + b_j$, and merge them if they fit the following criteria: $\mid m_i - m_j \mid < 0.25$, $\mid b_i - b_j \mid < 10$ pixels, and there are at least 10 pixels of L_i within a distance of 5 pixels of L_j.

From the initial set of candidate lines, we seek to find the subset of them that have the highest probability of being stair edges. We base our filtering techniques on the following assumptions: stair edges will be nearly parallel to each other (although they may be at any angle due to an oblique viewing direction); the lines will overlap a substantial amount in the horizontal direction (that is, they appear stacked vertically); and the points on the step edges lie on an inclined plane. To restrict our line set based on these assumptions, we first compute the angle in camera coordinates of each line and compute an angle histogram with 10° bins. We consider only the bin with the highest frequency but also keep lines that are within a -5° to $+5^\circ$ band around this bin; all lines with other orientations are removed.

To cluster the lines in the image and ensure that they are vertically localized, we compute a histogram that counts the number of lines that cross each column of the image. We compute the maximum bin value h_{max} in the histogram and find the widest rectangle, in image coordinates, such that all histogram bins in that range have frequencies of h_{max}. Any lines that do not fall within this region are rejected. We perform two more filtering steps before we confirm a hit with the detector. Since a pair of lines could constitute a texture or other physical artifact, we require at least three candidate lines in order to proceed. Lastly, we extract 3-D points from the remaining lines and fit a least-squares plane to them. We only accept the set as stair edge points and return a positive detection if the plane is at an angle with the horizontal that fits with the physical parameters of a traversable staircase (in our case, between 0° and 45°, covering the range of stairway pitches typically allowed by building codes). If a positive detection is confirmed at this step, then the set of extracted edge points is passed on to the stairwell modeler module.

---

4. Experiments

Our system has been tested extensively on data collected at a MOUT site on all of the available stairway types at the site, as well as on numerous negative examples. There were over 10 stairway types throughout the site, both indoor and outdoor (figure 5), of a variety of dimensions, and ranging from a few steps to a full flight. Two examples (shown in figures 6 and 7) had riser-less steps, and one type was partially curved. It has also been tested at a building at the UB. These datasets consist of nine recorded trials (seven and two, respectively).

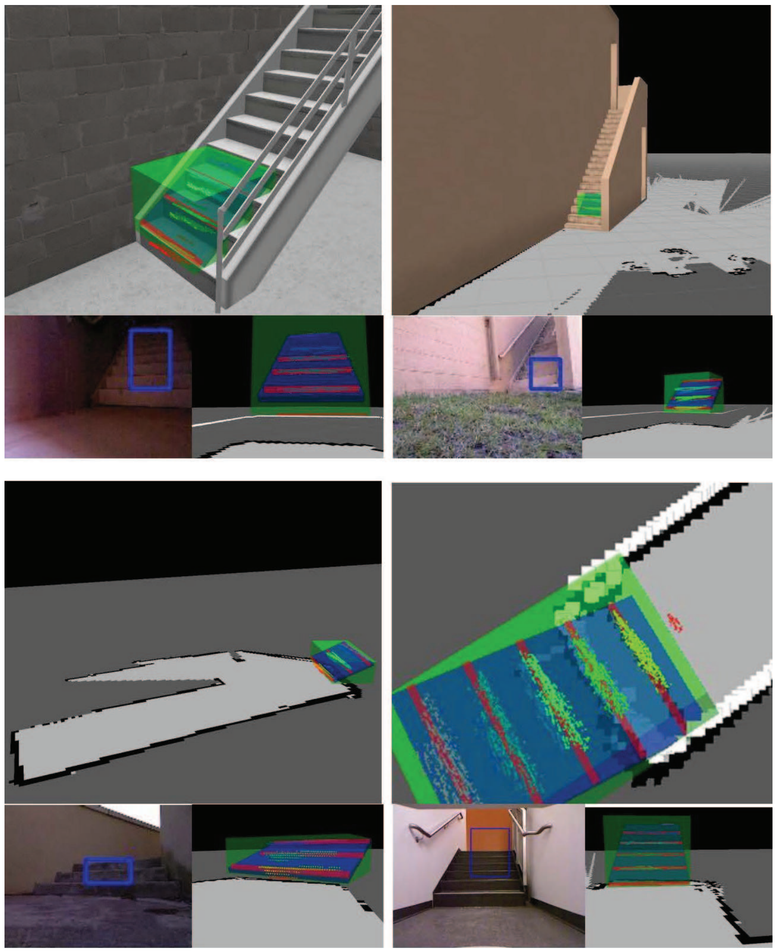

Figure 5. Results of several runs from our datasets: Building 1 Rear (top left), Building 3 (top right), Building 7 Exterior (bottom left), and Davis Hall Rear (bottom right).

Our experiments use an iRobot PackBot mounted with a Microsoft Kinect depth sensor for the MOUT trials, and a TurtleBot (also with a Kinect sensor) for the UB data. Our system is implemented in C++ in the Robot Operating System (ROS) environment, with image processing performed using OpenCV, and point cloud processing with the Point Cloud Library (PCL) (*18*). Although the Kinect restricts the usable range of the detector and limits outdoor use to shaded areas, the dense depth image it produces provides high quality input data for our system. Several of the trials from the MOUT dataset were captured outdoors and indicated good performance with

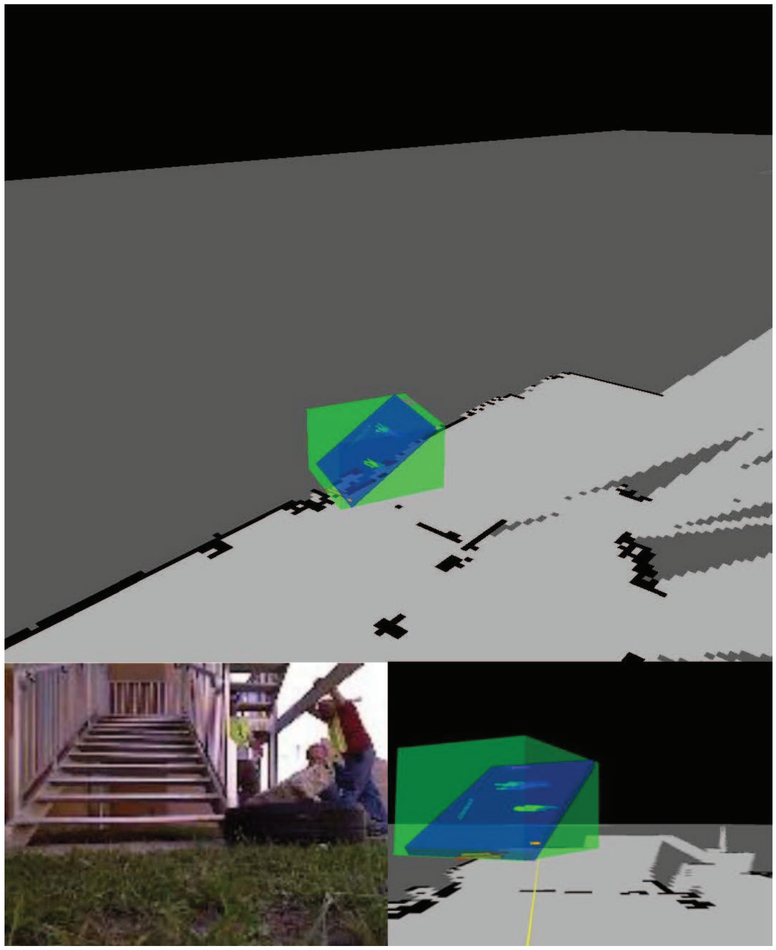

Figure 6. Figure showing one failure mode of our system, including camera and simulated views from the robot's perspective. Note that this stairway is both outdoors and of a riser-less style, two challenges for our implementation.

even somewhat compromised depth data. In principle, our approach could be applied out of the box to a depth map produced by a stereo camera for outdoor detection, but this is as yet untested.

4.1 Detection

The detector is robust to viewing angle, stair appearance and size, and even partially curved (although non-spiral) stairs. We obtained detections on 100% of the stairways on which we tested the system, and obtained enough observations to build a model for all of the indoor stairways we observed and all but one of the outdoor stairways (figure 6). However, one other outdoor stairway failed to converge to an accurate model (figure 7). These failure modes for outdoor stairways reveal a limitation of the sensor, as well as the challenge of scan alignment

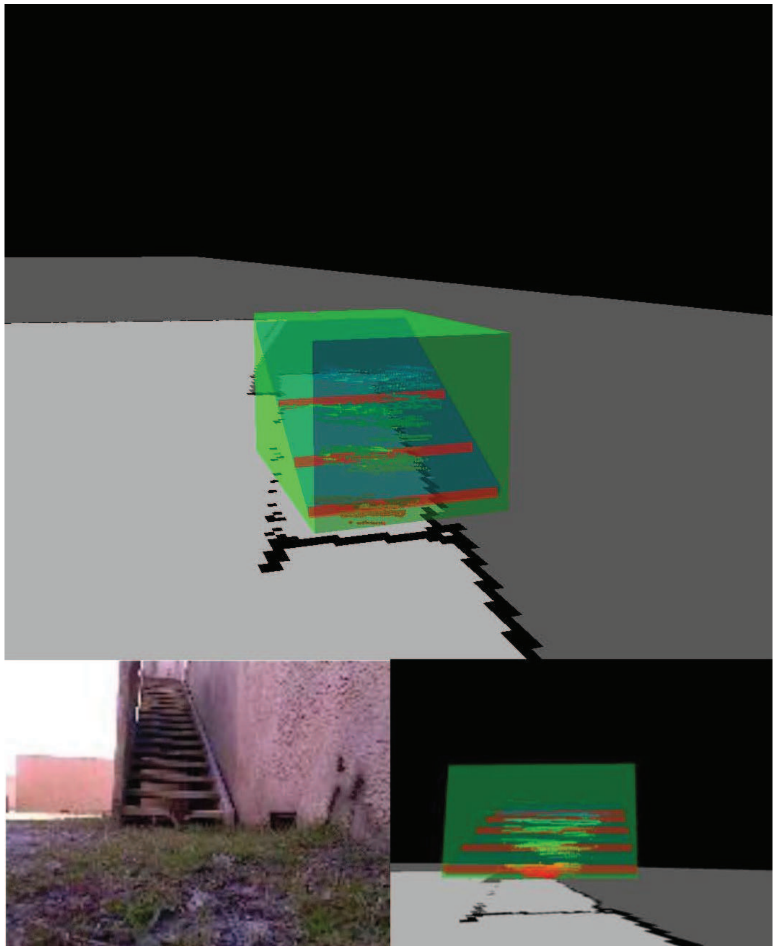

Figure 7. Figure showing another failure mode, including the final model for the Container trial and camera view from the robot's perspective. Although our method is fairly robust to misalignment, this example illustrates failure due in part to an uneven ground plane and its corresponding odometry challenges.

under noisy odometry. Over many hours of testing, there were a small number of false positive observations (occurring only from a stairway railing and a set of bleachers), but in no instances were there enough false detections to create a model when not in the presence of a stairway, and these were greatly in the minority when a stairway was present, resulting in the false observations being removed as statistical outliers and ultimately not affecting the model.

In all instances, the system requires less than 0.01 s to process each 320×240 frame on both an onboard Intel Core i7 and an offboard Mac Mini when run in post-processing. As it is extremely lightweight, the detector can effectively run in the background without requiring many of the resources needed for all of the other processes performing exploration in real time. Including the modeling step, the full system is able to operate at $20+$ Hz concurrent with mapping.

Another such failure mode exists if the ground surface is uneven, as with natural environments where our ground plane assumption does not hold. Figure 7 illustrates another example of a trial in which our approach failed to produce and accurate model. Here, the rough terrain causes misalignment of the extracted edges, and although the bounding box is localized fairly well, the alignment is off and the step estimates are imprecise.

4.2 Modeling

Visual results of modeling for all nine trials in the two datasets can be found throughout this report in figures 1, 2, 3, 5, 6, and 7. Where possible, rendered 3-D models of the corresponding buildings were superimposed and aligned with the map such that the stairway model is overlaid.

We also measured ground truth step dimensions and pitch for several of the trials, and we present those results in table 1. Each of these results was achieved with < 100 observations. In general, the estimates are quite accurate, modeling the step dimensions to within 2 cm and the pitch to within $3°$, on average. However, one frequent source of inaccuracy is underestimation of the step width, with a mean error of 17 cm. This is expected, though, based on the stair detection procedure, which only extracts edge points in a horizontal window of the depth image where all of the edge lines overlap, leading to observations that are always narrower than the lines producing them. Each trial's results indicate that a robotic platform would be able to determine the traversability of that stairway using this system.

We also present some results showing the convergence of the models for several trials. Figure 8 shows the evolution of the model parameters over time for the two UB trials. Here, all of the parameters are normalized by their ground-truth values, so each quantity should tend toward 1 over time. Both trials indicate that after a small number of detections, the models approach their final state.

Table 1. Table of model step estimates and ground truth values (GT).

Trial	Height (m)	Depth (m)	Width (m)	Pitch (°)
B.1 Front	0.174	0.257	0.977	34.0
GT	**0.196**	**0.254**	**0.965**	**37.6**
B.1 Rear	0.166	0.252	0.715	33.0
GT	**0.196**	**0.254**	**0.965**	**37.6**
B.3	0.169	0.245	0.689	35.4
GT	**0.192**	**0.263**	**1.015**	**36.2**
Davis Front	0.175	0.323	1.076	28.3
GT	**0.181**	**0.305**	**1.226**	**30.7**
Davis Rear	0.169	0.311	1.048	29.4
GT	**0.165**	**0.292**	**1.175**	**29.5**
Mean Error	0.017	0.012	0.173	2.3
Variance	0.000195	0.000246	0.0165	3.52

5. Future Work

Ultimately, we want this work to enable a new robot behavior: fully autonomous multi-floor exploration by ground robots. With the localization and modeling system presented here, we aim to take the first step in that direction. Other problems that would still need to be solved include incorporation of elevation measurements into both the mapping and exploration algorithms, execution of an autonomous stair climbing routine after a stairwell is found, and modification of path planning algorithms to set stair traversal as a high, but finite, cost path.

One immediate problem for our future work is the extension of our approach to multiple stairways within a single map. We currently assume that there is only one stairway in the environment, but in lifting that assumption, we will need the system to determine whether each new observation should contribute to an existing model, or whether it represents the detection of a new stairway in the environment. For all but very wide stairways, proximity should be a strong discriminator between edges extracted from different stairways, with such observations being much farther from an existing model than ordinary statistical outliers.

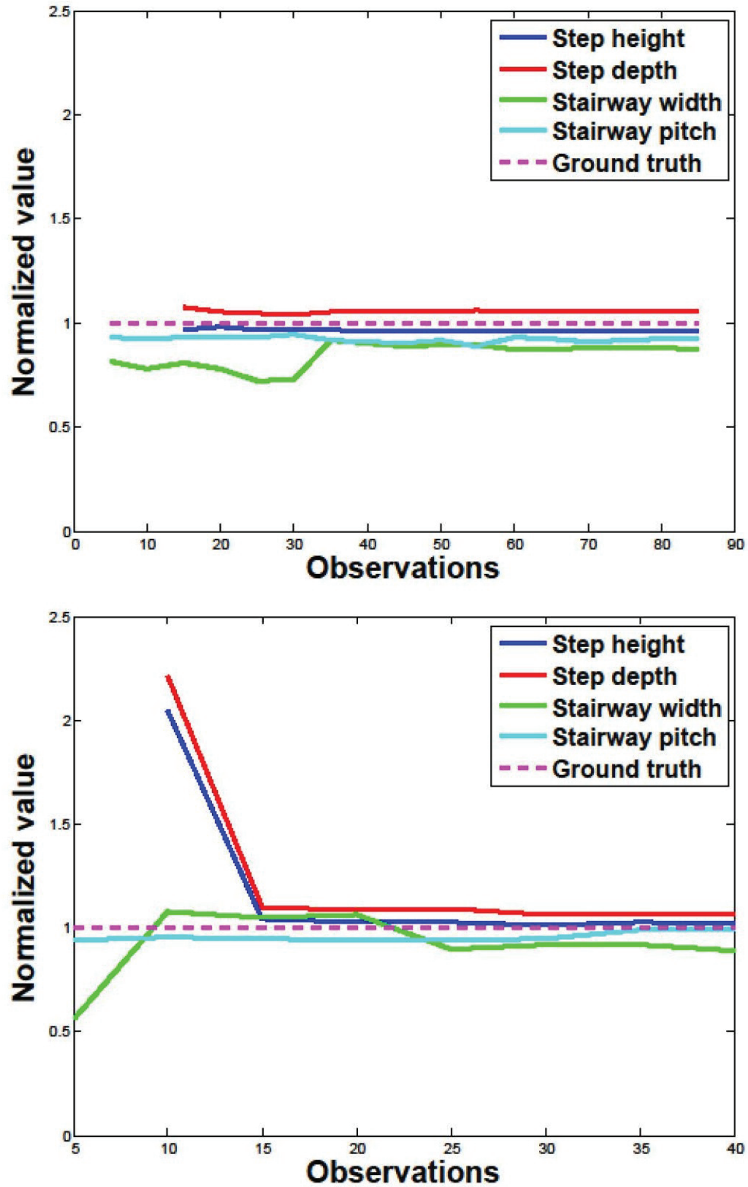

Figure 8. Convergence of normalized model parameters for Davis Front (top) and Davis Rear (bottom) trials from the UB dataset.

Another major component of a multi-floor exploration system is a solution to the complementary problem of descending stairway detection and modeling. Although there is no explicit modeling in reference 17, they present techniques for performing detection of descending stairways using cues from camera imagery. Our approach for ascending stairway detection is not transferable to the descending problem, but could be paired with such a system. Additionally, by maintaining models of any previously ascended stairways, a robot could more easily descend those stairways for which it has a location and model if the full dimensions of the stairway were determined during ascension, using the initial model as a base from which to extrapolate.

Another potential research avenue is the extraction of lines directly from the point cloud, rather than from the depth image. This would open up our modeling technique to a wider variety of 3-D sensors, and enable post-processing of 3-D maps for stair locations.

6. Conclusions

We present a novel, minimal, generative model for a set of stairs, as well as a system for fitting that model to data extracted and aggregated from many observations of a stairways with a depth camera. Our model is sufficiently detailed to permit the robot to determine the traversability of a set of stairs, while simple enough to be computed in real time and robust to errors. Providing the observations for the modeling module is a stair detector that uses image processing techniques to find lines of depth discontinuity and enforce geometric constraints on them in order to extract the points on just the lines corresponding to stair edges. We have tested our system on a variety of stairways in both indoor and outdoor environments, as well as in many environments where no stairs exist. The results indicate both robustness in detection and modeling, and accuracy in parameter estimation. This work represents an initial step toward autonomous multi-floor exploration by unmanned ground vehicles.

7. References

1. Johnson, A. M.; Hale, M. T.; Haynes, G. C.; Koditschek, D. E. Autonomous Legged Hill and Stairwell Ascent. in *IEEE International Workshop on Safety, Security, & Rescue Robotics (SSRR)*, November 2011, pp. 134–142.

2. Mihankhah, E.; Kalantari, A.; Aboosaeedan, E.; Taghirad, H.; Ali, S.; Moosavian, A. Autonomous Staircase Detection and Stair Climbing for a tracked Mobile Robot Using Fuzzy Controller. in *Proceedings of the IEEE International Conference on Robotics and Biomimetics. IEEE*, 2008, pp. 1980–1985.

3. Ray, R.; Bepari, B.; Bhaumik, S. On Design and Development of an Intelligent Mobile Robotic Vehicle for Stair-case Navigation. in *Intelligent Autonomous Systems. Springer*, 2010, pp. 87–122.

4. Osswald, S.; Gorog, A.; Hornung, A.; Bennewitz, M. Autonomous Climbing of Spiral Staircases with Humanoids. in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE*, 2011, pp. 4844–4849.

5. Pradeep, V.; Medioni, G.; Weiland, J. Piecewise Planar Modeling for Step Detection Using Stereo Vision. in *Workshop on Computer Vision Applications for the Visually Impaired*, 2008.

6. Lu, X.; Manduchi, R. Detection and Localization of Curbs and Stairways Using Stereo Vision. in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE*, 2005, pp. 4648–4654.

7. Cong, Y.; Li, X.; Liu, J.; Tang, Y. A Stairway Detection Algorithm Based on Vision for UGV Stair Climbing. in *IEEE International Conference on Networking, Sensing and Control. IEEE*, 2008, pp. 1806–1811.

8. Mourikis, A.; Trawny, N.; Roumeliotis, S.; Helmick, D.; Matthies, L. Autonomous stair Climbing for Tracked Vehicles. *The International Journal of Robotics Research* **2007**, *26* 7, 737.

9. Wang, S.; Wang, H. 2d Staircase Detection Using Real Adaboost. in *International Conference on Information, Communications and Signal Processing. IEEE*, 2009, pp. 1–5.

10. Hernandez, D.; Jo, K. Outdoor Stairway Segmentation Using Vertical Vanishing Point and Directional Filter. in *International Forum on Strategic Technology (IFOST). IEEE*, 2010, pp. 82–86.

11. Hernandez, D.; Kim, T.; Jo, K. Stairway Detection Based on Single Camera by Motion Stereo. *Modern Approaches in Applied Intelligence* **2011**, pp. 338–347.

12. Schmittwilken, J.; Saatkamp, J.; Forstner, W.; Kolbe, T.; Plumer, L. A Semantic Model of Stairs in Building Collars. *Photogrammetrie Fernerkundung Geoinformation* **2007**, *2007* (6), 415.

13. Schmittwilken, J.; PlÍumer, L. Model Selection for Composite Objects with Attribute Grammars. in *Proceedings of 12th AGILE Conference on GIScience*, 2009.

14. Shen, S.; Michael, N.; Kumar, V. Autonomous Multi-floor Indoor Navigation with a Computationally Constrained Mav. in *Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE*, 2011, pp. 20–25.

15. Ozkil, A.; Fan, Z.; Xiao, J.; Dawids, S.; Kristensen, J.; Christensen, K. Mapping of Multi-floor Buildings: A Barometric Approach. in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on. IEEE*, pp. 847–852.

16. Prieto-MaraŸnt'on, V.; Cabrera-Gt'amez, J.; Domt'inguez-Brito, A.; Hernandez-Sosa, D.; Isern-Gonzt'alez, J.; Fernt'andez-Perdomo, E. Efficient Plane Detection in Multilevel Surface Maps. *Journal of Physical Agents* **2011**, *5* (1), 15–23.

17. Hesch, J.; Mariottini, G.; Roumeliotis, S. Descending-stair Detection, Approach, and Traversal with an Autonomous Tracked Vehicle. in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE*, 2010, pp. 5525–5531.

18. Rusu, R. B.; Cousins, S. 3D is Here: Point Cloud Library (PCL). in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9–13 2011.

19. Fischler, M. A.; Bolles, R. C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* **1981**, *24* (6), 381–395.

20. Canny, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **1986**, (6), 679–698.

21. Kiryati, N.; Eldar, Y.; Bruckstein, A. A Probabilistic Hough Transform. *Pattern Recognition* **1991**, *24* (4), 303–316.

NO. OF COPIES	ORGANIZATION
1 (PDF ONLY)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FORT BELVOIR VA 22060-6218
1 HC	US ARMY RESEARCH LAB IMAL HRA 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	US ARMY RESEARCH LAB RDRL CIO LL 2800 POWDER MILL RD ADELPHI MD 20783-1197
10	US ARMY RESEARCH LAB JEFFREY DELMERICO JASON CORSO DOUGLASS SUMMERS-STAY STUART YOUNG JONATHAN FINK JOHN ROGERS ETHAN STUMP JASON GREGORY DAVID BARAN PHILIP DAVID

INTENTIONALLY LEFT BLANK.